Com-plete Files and Associated User Files

Maintenance of the Com-plete system requires knowledge of the files distributed with Com-plete, the files created during Com-plete installation, and user files associated with Com-plete. This chapter describes the functions of file allocation, file backup/recovery, and file compression relocation.

You should write off entire disk packs at periodic intervals so that specific files can be restored as needed. Restoring files to their original location preserves the validity of the VTOC and catalog entries. Depending on the specific requirements of your installation, additional backup/restore procedures may be necessary.

Standard operating system backup and recovery techniques can be used for VSAM, ISAM, and BDAM files.

This chapter covers the following topics:

- COMSD - Com-plete Sequential/Direct Dataset
- Dynamic SD files
- COMSPL - Com-plete Spool Data Set
- CAPTURn - Com-plete Capture File(s)
- COMSYSn - Com-plete System Data Containers
- LOAD - Distributed Load Module Library
- USER LOAD - User Load Module Library
- MAP Library
- PROFILES - Editor Profiles Library
- SOURCE - Com-plete Distributed Source Library
- UDEBUG Profile Library
- UDEBUG Text Card Library
- Edit Source Libraries
- Application-Specific Data Sets
- COMDMP Dump Data Set (VSE only)

COMSD - Com-plete Sequential/Direct Dataset

Use of COMSD, the Com-plete sequential/direct dataset is required.

DD/DLBL	COMSD
File name	COM.SD

Description

The Com-plete sequential/direct (SD) dataset is internally split into two sections - one for application SD files and paging files, and one for Com-plete online dumps. Separate directories exist for these two sections. The first record of the dataset containes central information about the dataset and its two sections.

As an alternative to allocating SD files within the COMSD dataset, you can choose the option to dynamically allocate each SD file as a separate VSAM relative record dataset. See the section **Dynamic SD Files** below.

Allocation

This dataset must be allocated as a VSAM Relative Record Dataset and initialized using the Com-plete BATCH utility TUSDUTIL before it can be used. When deciding on the size of this dataset, you must consider the maximum number and sizes of application SD and paging files that will be allocated and the space required for online dumps. For detailed information about allocation and initialization parameters, refer to the description of BATCH utility TUSDUTIL in this documentation.

A sample allocation and initialization job is provided on the installation tape. You can use this job as a model (see the *Com-plete Installation and Migration documentation*).

Compression

Free space is reused, therefore no compression is necessary.

Backup/Restore

Backup and restoration of the entire SD dataset using VSAM utilities is possible only when Com-plete is not active. Selective restoration of SD files from this kind of BACKUP is not supported. TUSDUTIL can be used to backup SD files with optional deletion of "old" SD files and for full or selective restoration of SD files.

Backup and restore of SD files using this utility can be performed only while Com-plete is active with SYSPARM ACCESS=YES. TUSDUTIL supports restoration of SD files from V44 and V45 BACKUP datasets written by the former Com-plete BATCH utility TUSRSDCM.

Reallocation

There are no restrictions on reallocation of the SD dataset.

Dynamic SD files

Requires DFSMS 1.3 or higher.

As an alternative to allocating SD files within the COMSD dataset, you can choose the option to dynamically allocate each SD file as a separate VSAM relative record dataset. If you opt to do so, COMSD is still required, but is used for online dumps only. In this case, in order to avoid wasting disk space, SDFILES=0 should be specified when initializing COMSD.

DD name: SYSnnnnn (generated dynamically during allocation)

File name:

prefix.$. tempname .T tid	for temporary SD files,
prefix.STIMER. &sysname	for Com-plete’s UTIMER SD file,
prefix. name .T tid	in all other cases.

Where:

prefix	is the dataset name prefix specified by SYSPARM SDPREFIX,
tempname	is the name of the temporary SD file with the leading ’&&’ omitted,
tid	is the value of the TID parameter of the SDOPEN function, converted into a fixed-length five character presentation (SHR is represented as 00000),
&sysname	is the value of the system variable &sysname,
name	is the value of the NAME parameter of the SDOPEN function.

Description

SD files are allocated dynamically as separate VSAM datasets when the SYSPARM SDPREFIX is present, otherwise COMSD will be used.

The advantages of dynamic SD files are:

the ability to share SD files between multiple instances of Com-plete, e.g., in a Parallel Sysplex. The COMSD dataset cannot be shared between multiple instances of Com-plete.

The ability to use standard SMS features for backup, migration, restore, etc. of each single SD file.

Special caution is required when running multiple Com-pletes using the same SDPREFIX value, for two reasons:

1. Applications sharing SD files might require changes in the serialization mechanism they use, e.g., the scope of an ENQ might have to be changed from ’step’ into ’systems’.

2. Applications (e.g., Com-plete’s editor UEDIT) might not expect the same SD file to be opened more than once at a time. Unpredictable results including abnormal program termination and loss of data may occur when the same userID invokes multiple instances of such an application at the same time from different systems. The only secure way to use this configuration is when all Com-pletes sharing an

SD file prefix are entered through the same Generic VTAM Resource name, thus ensuring uniqueness of each session.

Allocation

You don’t have to allocate dynamic SD files, they are allocated automatically by Com-plete during execution of an SDOPEN function and deleted during execution of an SDDEL function.

The parameters Com-plete uses when allocating an SD file depend on the following:

The values of the RECORDS and RECLEN parameters of the SDOPEN function. These values are used to calculate the amount of space required.

The DFSMS DATACLASS, STORAGECLASS, and MANAGEMENTCLASS specified for SYSPARM SDSMSCLASS. You might wish to create extra classes to be used for SD files. Software AG recommends that you define only the absolute minimum of parameters for DATACLASS and STORAGECLASS, allowing DFSMS to choose optimum values for CONTROLINTERVALSIZE, etc.

To minimize system affinity in a Parallel Sysplex, SD file access should make use of the Record Level Sharing (RLS) option. To enable Com-plete to exploit this feature, LOG=NONE must be specified for the STORAGECLASS and Com-plete SYSPARM SDRLS=YES must be specified. Note that the LOG parameter of the STORAGECLASS is available only starting from DFSMS 1.4. For DFSMS 1.3, the DATACLASS parameter SHAREOPTIONS=(3,3) or (4,3) should be specified instead, and SYSPARM SDRLS should be omitted.

Backup/Restore/Reallocation

Com-plete does not provide any special utilities for nor does it put any restrictions on backup, restore, and reallocation of dynamic SD files.

Security considerations

Dynamic SD files are always created, opened, and deleted with Com-plete’s ACEE in effect, never with the ACEE of the user. Therefore, authorization should be defined similar to other Com-plete datasets.

COMSPL - Com-plete Spool Data Set

Use of COMSPL, the Com-plete message switching/printout spooling file, is required.

DD/DLBL	COMSPL
File Name	COM.SPOOL

Description

The Com-plete spool data set contains messages and printout spool files written by Com-plete or by user applications.

Allocation

You must allocate the data set as a VSAM Relative Record Data Set ("RRDS"). Due to VSAM restrictions, the record size must be 7 bytes smaller than the CISIZE of the VSAM dataset. The minimum CISIZE for this data set is 512 (record size 505) and the maximum size is 8192 (record size 8185). When deciding on the size of this data set, you must consider the general size of the printouts that will be written to it: the larger the printouts, the larger the record size should be. It must then be initialized with the INIT function of the TUMSUTIL utility with the number of records that it should contain. See the description of the TUMSUTIL utility in the chapter Batch Utility Programs for more information.

The size of the spool data set is highly dependent on your installation’s usage of message switching and printout spooling. The standard installation job allocates a 5-cylinder 3380 data set, but you will probably need to enlarge this. Please be aware that the spool data set must contain AT LEAST as many records as the value specified for the maximum printout sysparm MAXPRINTOUT (see the description of this parameter). A sample allocation and initialization job is provided on the installation tape. You can use this job as a model (see the Com-plete Installation documentation).

Compression

Free space is reused. No utility is therefore necessary for the compression of the data set.

Backup / Restore

The entire spool data set can be backed up and restored using standard IDCAMS utilities. The TUMSUTIL utility enables the installation to backup and restore printouts to the data set on a selective basis.

Reallocation

The Com-plete spool data set can be reallocated and moved at will without affecting other Com-plete data sets.

CAPTURn - Com-plete Capture File(s)

Use of the Com-plete capture file(s) is optional.

DD/DLBL	CAPTURn
File Name	COM.CAPTUR

Description

The Com-plete capture data sets are where Com-plete writes all its capture information. Only one data set is in use at any one time: when this data set is full, Capture will begin using another data set, and the full data set is freed for use.

You can define up to nine Capture data sets to Com-plete: CAPTUR1, CAPTUR2.....CAPTUR9. To enable the Capture facility, the sysparm CAPTURE must also be specified (see the chapter Initialization for more information).

Allocation

Capture data sets are allocated as VSAM Entry Sequenced Data Sets (ESDS) with a variable record size of between 32 and 4096 bytes. Prior to use, the Capture data sets must also be initialized using the TUSACAPT utility. This utility is described in the chapter Batch Utility Programs. A sample job to allocate two capture data sets is provided on the distributed source data set (see the Com-plete Installation documentation).

Compression

No compression of the Capture data sets is necessary, as the data is normally written to the data set and copied off, or overwritten by more recent Capture data.

Backup/Restore

It is not necessary to backup or restore Capture data sets in the normal way, because these data sets are normally copied and reinitialized. You can use standard IDCAMS utilities to copy the data from these data sets and reinitialize them using TUSACAPT.

COMSYSn - Com-plete System Data Containers

Use of COMSYSn, the Com-plete System Data Set, is required.

DD/DLBL	COMSYS1	File name	COM.VSAM.SYSTEM.BASE
	COMSYS3		COM.VSAM.SYSTEM.USERDEF
	COMSYS4		COM.VSAM.SYSTEM.CATALOG

Description

The Com-plete System Data Containers hold most of the information describing the environment of one or more Com-plete’s. Examples are the Com-plete system messages, the User ID definitions (logon definition plus user profiles, PF keys etc.).

Allocation

Allocate COMSYSn via the VSAM utility IDCAMS function DEFINE. Some initial data is loaded from the distribution tape via IDCAMS REPRO. These data sets are allocated and initialized during the standard installation process. You must review the size of these files based on the amount of installation data that exists on the files. A sample job is supplied on the distributed source data set to allocate and initialize these data sets. Please refer to the Com-plete Installation documentation for more details.

Compression, Backup/Restore

Standard VSAM methods (using IDCAMS) can be used for these purposes; therefore, no Com-plete utility is provided for these functions.

Reallocation

Can also be done via IDCAMS as required.

LOAD - Distributed Load Module Library

Use of LOAD, Com-plete's distributed load module library, is required.

DD/DLBL	COMPLIB/In LIBDEF PHASE,SEARCH=
File Name	COM.LOAD

Description

The Com-plete distributed load library contains the executable load modules, tables, and maps, including the Com-plete nucleus. This file should be the second file in the VSE LIBDEF SEARCH string. Please refer to **Initialization** in this documentation for the data set order under MVS.

Allocation

The Com-plete load library is a standard MVS load module library or a standard VSE library. Please refer to the Com-plete Installation documentation for the necessary space allocation for this data set.

In MVS, the load library contains all single CSECTS required to relink any Com-plete program, as well as all programs dynamically loaded during initialization.

In VSE, this library contains all modules with MEMBERTYPE PHASE and OBJ required to relink any Com-plete program, as well as all programs loaded dynamically during initialization.

Compression, Backup/Restore

Use standard operating system utilities to backup and restore these libraries.

Reallocation

There are no restrictions on reallocation of the load library.

USER LOAD - User Load Module Library

Use of USER LOAD, Com-plete's distributed user load module library, is optional, depending on whether installation-specific load modules are required.

DD/DLBL	COMPLIB/In LIBDEF PHASE,SEARCH= and LIBDEF PHASE,CATALOG=
File Name	COM.USER.LOAD

Description

The Com-plete user load library is allocated as an empty library during installation. The user load library should contain the executable load modules, user exits, tables, and maps, including the Com-plete control program that has been specifically tailored for your installation. This file should be the first file in the VSE LIBDEF SEARCH string or the MVS COMPLIB concatenation.

In VSE, this is the designated map library for UMAP.

Allocation

The Com-plete user load library is a standard MVS load module library or a standard VSE library. During standard installation, a data set of 10 3380-type cylinders is allocated. This should be reviewed based on installation usage.

Compression, Backup/Restore

Use standard operating system backup/restore utilities for this library.

Reallocation

There are no restrictions on reallocation of the distributed user load module library.

MAP Library

Use of the Com-plete map library is required for the UMAP utility.

DD	`SYSMAP:   VSE LIBDEF PHASE,CATALOG=`
File Name	COM.MAPS

Description

The UMAP online utility saves and retrieves maps from the map library.

The Com-plete map library is allocated as an empty library during installation.

In MVS, the map library must also be in the COMPLIB concatenation so that maps created by UMAP can be used by programs.

Allocation

The standard installation allocates 2 cylinders of 3380-type devices for maps.

MVS	The MAP library is a standard MVS load module library.
VSE	The maps are placed in the user load library.

Compression, Backup/Restore

Use standard operating system backup/restore utilities for this library.

Reallocation

There are no restrictions on reallocation of the map library.

PROFILES - Editor Profiles Library

Use of Com-plete's editor profiles library is optional.

Description

The User Profiles used by the editor are created and maintained in the Profile library.

The Profile library is the first source statement library found in either the terminal operator's library ID table or the Com-plete UEDTB1 table that is defined with "$$" as the two-character ID. No library is specifically allocated during installation.

Allocation

The Profile library is a standard MVS library with LRECL=80, or a standard VSE library. Because the library is opened each time a Profile is accessed/updated, secondary allocation is allowed.

Each Profile member is roughly 1 to 35 card images. Be sure to provide sufficient directory entries for the users who are expected to use the editor.

In MVS, allocate with RECFM=FB, DSORG=PO, and LRECL=80.

Compression, Backup/Restore

Use standard operating system backup/restore utilities for this library.

Reallocation

There are no restrictions on reallocation of the editor profiles library.

SOURCE - Com-plete Distributed Source Library

DD/DLBL	Not applicable
File Name	COM.SOURCE

Description

The Com-plete distributed source library is a standard MVS source library, or a standard VSE library. You are recommended to leave the members on this library unchanged for reference purposes. When a member must be changed, it can be copied to the user source. The library contains:

- Macros and/or VSE Edecks necessary to assemble user exits or Assembler applications;
- COBOL COPY code and PL/I %INCLUDE code necessary to compile applications in COBOL and PL/I;

- Macros and/or VSE Edecks required to assemble TIBTAB;
- Sample job control and data required to perform Com-plete system maintenance and establish the desired user-written security routines;
- The CMOSTYPE macro/Edeck required by some Com-plete macros.

Allocation

The source library is a standard MVS library with LRECL=80, or a standard VSE library. Secondary allocation is allowed.

This data set is allocated during standard installation, see the Com-plete Installation documentation for more information.

Compression, Backup/Restore

Use standard operating system backup/restore utilities for this library.

Reallocation

There are no restrictions on reallocation of the distributed source library.

UDEBUG Profile Library

The use of the UDEBUG facility is optional.

DLBL	SAGLIB - SUBLIB=COMDBPRF
DD	COMDBPRF
File Name	Not applicable

Description

This dataset contains UDEBUG profiles which can contain UDEBUG commands to enable users to customize their UDEBUG session and which are read when the PROFILE command is issued and at startup for each individual user.

This DD name must point to a Partitioned Dataset with a fixed block record format. It must have a logical record length of 80 and can have any blocksize which is a multiple of 80. The blocksize of the dataset will have a direct impact on the catalog size for UDEBUG and therefore should be taken into account.

Allocation

The source library is a standard MVS library with LRECL=80, or a standard VSE library. Secondary allocation is allowed.

Compression, Backup/Restore

Use standard operating system backup/restore utilities for this library.

Reallocation

There are no restrictions on reallocation of the distributed source library.

UDEBUG Text Card Library

The use of the UDEBUG facility is optional.

DLBL	SAGLIB - SUBLIB=COMDBTXT
DD	COMDBTXT
File Name	Not applicable

Description

This dataset contains text cards generated by the assembler option TEST which can be subsequently read by the UDEBUG READ command to build symbols.

This DD name must point to a Partitioned Dataset with a fixed block record format. If must have a logical record length of 80 and can have any blocksize which is a multiple of 80. The blocksize of the dataset will have a direct impact on the catalog size for UDEBUG and therefore should be taken into account.

Allocation

The source library is a standard MVS library with LRECL=80, or a standard VSE library. Secondary allocation is allowed.

Compression, Backup/Restore

Use standard operating system backup/restore utilities for this library.

Reallocation

There are no restrictions on reallocation of the distributed source library.

Edit Source Libraries

Edit source libraries are standard operating system libraries, but their use by Com-plete requires special considerations and the creation of proper interfaces. This section describes the maintenance considerations of these libraries from a Com-plete point of view and explains how to create additional libraries and define the required interfaces.

Overview

Since edit source libraries are standard operating system libraries, their creation and maintenance has much in common with other operating system libraries not used with Com-plete. There are, however, additional considerations due to the implementation of an online environment and the requirements of Com-plete.

Com-plete is distributed with two online text editors:

- UED, the text editor designed for use with hard copy terminals;
- UEDIT, the text editor designed for use with 3270-type terminals or compatible devices.

The files to be edited by each of these text editors are not necessarily the same. Considerations for each editor are summarized in the following text. Additional details about each editor can be found in the Com-plete Utilities documentation.

Creation of Edit Source Libraries

The creation and implementation of edit source libraries consists of:

- Allocation of the libraries;
- Adding a UEDTB1 entry or using the UL function of the UUTIL utility.

Allocation

Edit source libraries are standard operating system libraries. In MVS, the allocation of these libraries limits which Com-plete text editor can be used. The possibilities are:

- UEDIT - Partitioned files, sequential files, LIBRARIAN libraries, and PANVALET libraries

  RECFM=F or RECFM=FB
  LRECL larger than 80
  LRECL less than 256

- UED - Partitioned or sequential files

  RECFM=F or RECFM=FB
  LRECL larger than 2
  LRECL less than 247

In either situation, the BLKSIZE of the files is restricted only by the thread size in which the respective text editor is to execute. If a large BLKSIZE is specified for a given edit file, it may be necessary to modify the catalog entry for the text editor in use to a larger region size. For the UEDIT utility, this means modifying the catalog entry for the modules UEBP and UEPDIN; for UED, the catalog entry for UED itself should also be modified.

In VSE, editor source libraries are allocated and formatted with LIBR.

Note that space allocations must be contiguous.

UEDTB1

Source libraries edited by UED and UEDIT can be referenced by name or by a two-character library identification code. UEDTB1 is a module of non-executable code used to define the two-character identification codes assigned to edit libraries. This module is loaded by both UED and UEDIT, and used only when a two-character library identification code is entered to reference a given library. This module is also loaded and used by the utility program UPDS/USERV. If a two-character code is referenced that is not in this table, an invalid request results. UEDTB1 is system-wide in scope and is shared by all Com-plete users.

A full description of the use, creation, and maintenance of the UEDTB1 module is given in the chapter Security and User Exit Facilities.

DD/DLBL statements are not used to allocate edit libraries except in VSE. All edit libraries are allocated dynamically when they are read and when a SAVE or SUBMIT request is issued. In VSE, the library may be allocated with a DLBL and EXTENT, or Com-plete will optionally construct a DLBL and EXTENT from information provided in UEDTB1 or UUTIL-UL.

Note:
A VSAM VSE library must be defined via UEDTB1 or UUTIL-UL.

UUTIL-UL

Each Com-plete user can define up to 24 personal two-character library identification codes using the online utility UUTIL, function UL. Note that each user’s library definitions are searched before the system-wide UEDTB1 definitions.

Note:
A VSAM VSE library must be defined via UEDTB1 or UUTIL.

For more information, refer to **UUTIL - User Environment Definition Utility** in the Com-plete Utilities documentation.

Maintenance of Edit Source Libraries

Since edit source libraries are standard operating system files, their maintenance consists of standard techniques used to maintain source libraries: compression for partitioned files, and reallocation/expansion for sequential files. For example, the MVS utility program IEBCOPY can be used to compress a partitioned file either in place or with an unload followed by a compression in place.

Protection

The utility programs UED and UEDIT perform a DYNALLOC on any source library into which a SAVE/READ function is being performed. This causes an ENQ to be obtained on the queue name SYSDSN. If the operation is a SAVE, then the ENQ is established as with DISP=OLD. For a READ function, the ENQ is established as with DISP=SHR. If another terminal user is actively reading or writing the same source library to which a SAVE or READ operation has been directed, a warning message is given, and the terminal operator has the option of waiting, terminating, or retrying. This same convention exists if a batch job that accesses the library being edited is executing; the terminal operator will be prevented from performing a SAVE operation in a library being accessed by another user.

Note:
In MVS, protection as provided by the TSO/ISPF Editor via the queue name ISPEDIT is not supported.

Application-Specific Data Sets

Application programs running with Com-plete can access VSAM data sets as described in the chapter Software Interfaces of this documentation. In MVS, BDAM and ISAM data sets can be accessed using special Com-plete functions.

Declaration to Com-plete

Application programs refer to DD/DLBL names. All DD/DLBL names referenced by application programs must be declared ("cataloged") to Com-plete using the FM function of online utility UUTIL. This declaration includes the data set name, disposition (MVS only), the name of the VSAM user catalog (VSE only), and other information.

For a detailed description of the FM function of the UUTIL utility, refer to the *Com-plete Utilities documentation*.

Allocation / Deallocation

The data set is allocated to Com-plete dynamically when an OPEN request is issued against the appropriate DD/DLBL name. The data set is deallocated when it is closed explicitly using the CLOSE or BATCH subfunctions of UUTIL-FM.

Compared to permanent allocation of data sets to Com-plete, this mechanism provides maximum flexibility of data set access by BATCH jobs and for data set maintenance (backup, restore, reallocation, rename, etc.), without needing to restart Com-plete.

For compatibility reasons, it is also possible to allocate data sets to Com-plete permanently by specifying DD/DLBL statements in the startup JCL procedure.

Note:
(MVS only): While Com-plete is active, permanently allocated data sets are available for access by BATCH jobs and for maintenance only with the restrictions defined by the JCL DISP parameter and by the SHAREOPTIONS attribute. Setting a permanently allocated dataset to BATCH status closes the dataset and disables online access, but neither deallocates the dataset, nor frees the SYSDSN ENQ held.

COMDMP Dump Data Set (VSE only)

The use of COMDMP, the Com-plete dump data set, is required.

DLBL	COMDMP
File	COM.VSAM.DUMPFILE

Description

The Com-plete dump data set contains a dump of all relevant storage areas in case of a Com-plete abend.

Allocation

The dump data set is VSAM RRDS (relative record dataset) with a required record length of 4080 bytes. The size depends on the size of the Com-plete partition. At least 30 cylinders are required for a 16M partition.

A sample job to allocate a dump data set is provided on the distributed source data set.

Compression

No compression of the dump data set is necessary, as the data is normally written to the data set and copied off, or overwritten by more recent dumps.

Backup/Restore

You can backup and restore the data set using standard IDCAMS functions.

Reallocation

The Com-plete dump data set can be reallocated and moved without any restrictions.